ORGANIZATION OF MAGIC II ADVANCED COMPUTER FOR AIRBORNE GUIDANCE SYSTEMS

THE ELECTRONICS DIVISION OF GENERAL MOTORS CORPORATION

MILWAUKEE 1, WISCONSIN

ORGANIZATION OF MAGIC II
ADVANCED COMPUTER
FOR AIRBORNE GUIDANCE SYSTEMS

21 November 1963

AC SPARK PLUG
The Electronics Division
of
General Motors Corporation

SECTION I

INTRODUCTION

The MAGIC family of miniature airborne digital computers has been developed at AC Spark Plug Division to fulfill the computational requirements for complete navigation, guidance and control of rocket-launched space vehicles, ballistic missile and aircraft systems. The MAGIC family is characterized by great flexibility and high reliability coupled with low weight, power and volume. MAGIC computers are all program controlled. Numerical quantities are represented as binary whole numbers, and arithmetic is performed serially with two's complement representation of negative numbers. MAGIC I is the prototype of the family and was the first complete airborne computer to have its logic functions mechanized exclusively with integrated circuits. The circuits used were Fairchild micrologic. Welded electronic encapsulated modules of conventional miniature components were used for those circuit functions which could not be implemented by micrologic. The memory is a novel 4096 word toroidal core memory which operates in a serial mode. Table 1 is a summary of the MAGIC I characteristics. This computer has been in operation in the AC Spark Plug laboratory for more than 6200 hours and has been used extensively in system tests with a breadboard Inertial Measurement Unit. This operating experience has shown that the original design goals, particularly those of reliability and flexibility have been achieved.

MAGIC II has the same general organization as MAGIC I but differs from MAGIC I in three important aspects. More complex logic circuits have been employed resulting in a smaller number of total circuits. The memory is composed of 4096 words of nondestructive (NDRO) program storage and 256 words of scratch pad or temporary storage (DRO). This memory operates twice as fast as the MAGIC I memory, resulting in a significant increase in overall computer speed. Micromodules have been employed for the circuit functions which could not be mechanized by micrologic. In general, MAGIC II has a smaller weight, power and volume than MAGIC I, but provides increased computational capability. A summary of MAGIC II characteristics is shown in Table 2.

Section II is a functional description of the MAGIC II computer organization. The order code is described in detail in Section III.

Type	Program controlled Whole binary numbers
Instructions	21 Single Address Two per word
Addressing	Direct Indirect Relative
Arithmetic	Sign + 23 fractional bits Two's complement
Speed	70 μsec short operations (add, subtract, shift, etc.) 258 μsec multiply (48 bit product) 398 μsec divide (48 bit dividend) Times include instruction and one operand access
Input-Output	Inertial Measurement Unit Star Scanner Autopilot Missile Control Operator Controls
Memory	4096 words Serial organization 4 μsec cycle 2 bits/cycle
Physical	35 lbs . 65 cubic feet 90 watts

Integrated Circuits		Conventional Components	
Buffers	116	Transistors	495
Counter Adapters	7	Diodes	2475
Flip-flops	82	Resistors	1623
Gates	1187	Capacitors	265
Half Adders	80	Transformers	72
Half Shift	626		
Total	2098	Total	4930

Table 1. MAGIC I Characteristics

Type	Program controlled Whole binary numbers
Instructions	22 Single Address Two per word
Addressing	Direct Indirect Relative
Arithmetic	Sign + 23 fractional bits Two's complement
Speed	38 μsec short operations (add, subtract, store, etc.) 190 μsec multiply (48 bit product) 342 μsec divide (48 bit dividend) Times include instruction and one operand access
Input-Output	Inertial Measurement Unit Fix-taking Unit Pilot's Display Navigation Display Operator Controls
Memory	4096 NDRO words 256 DRO words 4 μsec cycle 4 bits/cycle
Physical	26 lbs .50 cubic feet 80 watts

Integrated Circuits		Micro-components	
Buffers	94	Transistors	319
Counter Adapters	10	Diodes	520
Double Gates	526	Resistors	319
Half Adders	61	Capacitors	86
Half Shift	628	Transformers	120
Total	1319	Total	1364

Table 2. MAGIC II Characteristics

SECTION II

COMPUTER ORGANIZATION

The basic organization of the MAGIC II Digital Computer Assembly is shown in Figure 1. The registers that play a functional role in controlling the flow of information in the machine are shown in this figure. For simplicity, registers used to assure proper timing of various actions in the computer have been omitted from the diagram. Two kinds of information, instruction and data words, are used in the computer. This section describes the manner in which this information is processed.

Instruction words are always read from the 4096 word NDRO memory and directed to the instruction buffer register in the instruction processing unit. Each instruction word is composed of two 12-bit instructions; the two instructions are executed in sequence. For instructions requiring access to memory, the address portion of the instruction is routed to the address register. For certain variations of these instructions, the contents of the bias register or the instruction counter are added to the address portion of the instruction to form the effective address. If the effective address of the operand lies between 0 and 127_{10} or 3968_{10} and 4095_{10}, the first or second 128 words, respectively of the DRO memory are selected for reading or writing. Effective addresses between 128 and 3967 will select the NDRO memory and instructions or constant operands may be read out. Only instruction words may be read from locations 0 to 127_{10} and 3968_{10} to 4095_{10} of the NDRO memory. Writing in the NDRO memory can only be effected through the ground support equipment.

For shift instructions and certain other types of operations, the address portion is sent to the control counter instead of to the address register for further decoding. The contents of the instruction counter and bias register can be replaced with information from memory for certain kinds of instructions.

Numbers involved in arithmetic computations are read from memory and transferred serially to the arithmetic unit. There each number is routed to the appropriate one of three arithmetic registers. When results of arithmetic operations are stored in memory, they are transferred and stored serially, four bits at a time, at the address specified by the address register only if the address specifies a word in the DRO memory.

All computer input and output data words are directed to and from the input-output unit via the A register. Information transfer between the A register and memory for purposes of input or output requires an additional step. Routing of information to or from the specific part of the input-output unit is accomplished in accordance with the address field of the input or output instruction.

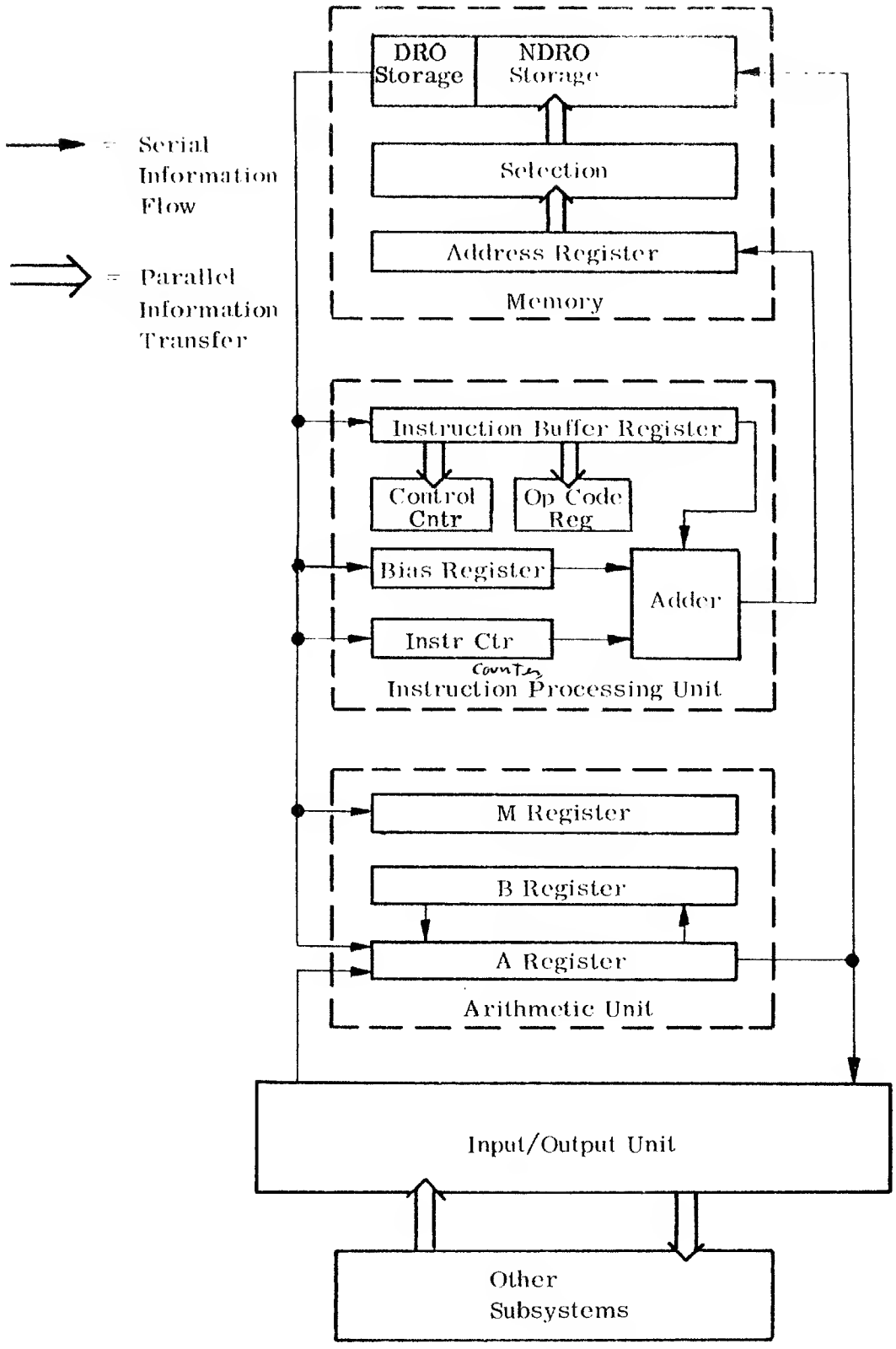


Figure 1. Simplified Block Diagram of the Computer Organization

Transfers between registers, or between memory and the rest of the computer, are generally made in a serial mode. Within the instruction processing unit and the input-output unit, information is transmitted 1 bit at a time, at a 1-megacycle rate. Within the arithmetic unit, information is transmitted 2 bits at a time, at either 1 Mc or 500 kc clock rates.

Information is transferred in and out of memory 4 bits at a time, at a 250 kc rate. To avoid inadvertent loss of information in the NDRO memory portion, the "write" electronics are not included as part of the airborne computer package.

The functional subsystems of this computer package are discussed in the following paragraphs.

(a) Instruction Processing Unit (IPU). The IPU provides the means for executing the various computer instructions in the order dictated by the computer program. A block diagram of the information flow is shown in Figure 2.

To minimize storage requirements for the program, special arrangements are used to obtain efficient use of the instruction bits. If fully random access were provided to the entire memory, the address field would need to be greater than 12 bits. Instead, a shortened address field with a number of variations in its meaning is used. By this means, an instruction containing both an operation code and an address can be represented by only 12 bits, allowing two instructions in a single 24-bit word.

Before discussing operation of the IPU, the format of the instructions and the various ways in which the address field can be interpreted will be described. Depending upon the type of instruction, either four or six bits are used to specify the operation code. The remaining bits are, in most cases, used to define a memory address. For certain instructions, however, the remaining bits are used for other purposes. The format of an instruction word for different types of operations is shown in Figure 3.

Two kinds of instructions involve reference to memory. Transfer instructions require reference to a memory table, in either the permanent or temporary memory, for the location of the next instruction. Core reference instructions require reference to memory to obtain or store data. For these two types of instructions, the instruction address field is used to generate an effective address for memory reference.

Four of the five transfer-type instructions use an eight-bit address field for indirect addressing. These transfer instructions use the address field to denote which word of the 256-word transfer table in the memory contains the address of the next instruction to be executed. The low-order 12 bits of the transfer table word are used as the next instruction address. The high-order 12 bits are sent to the bias register. The transfer table occupies locations 000_{10} to 255_{10}. Locations 000_{10} to 127_{10} are contained in the DRO memory and 128_{10} to 255_{10} are contained in the NDRO memory. The format of the transfer instruction is shown in Figure 3.

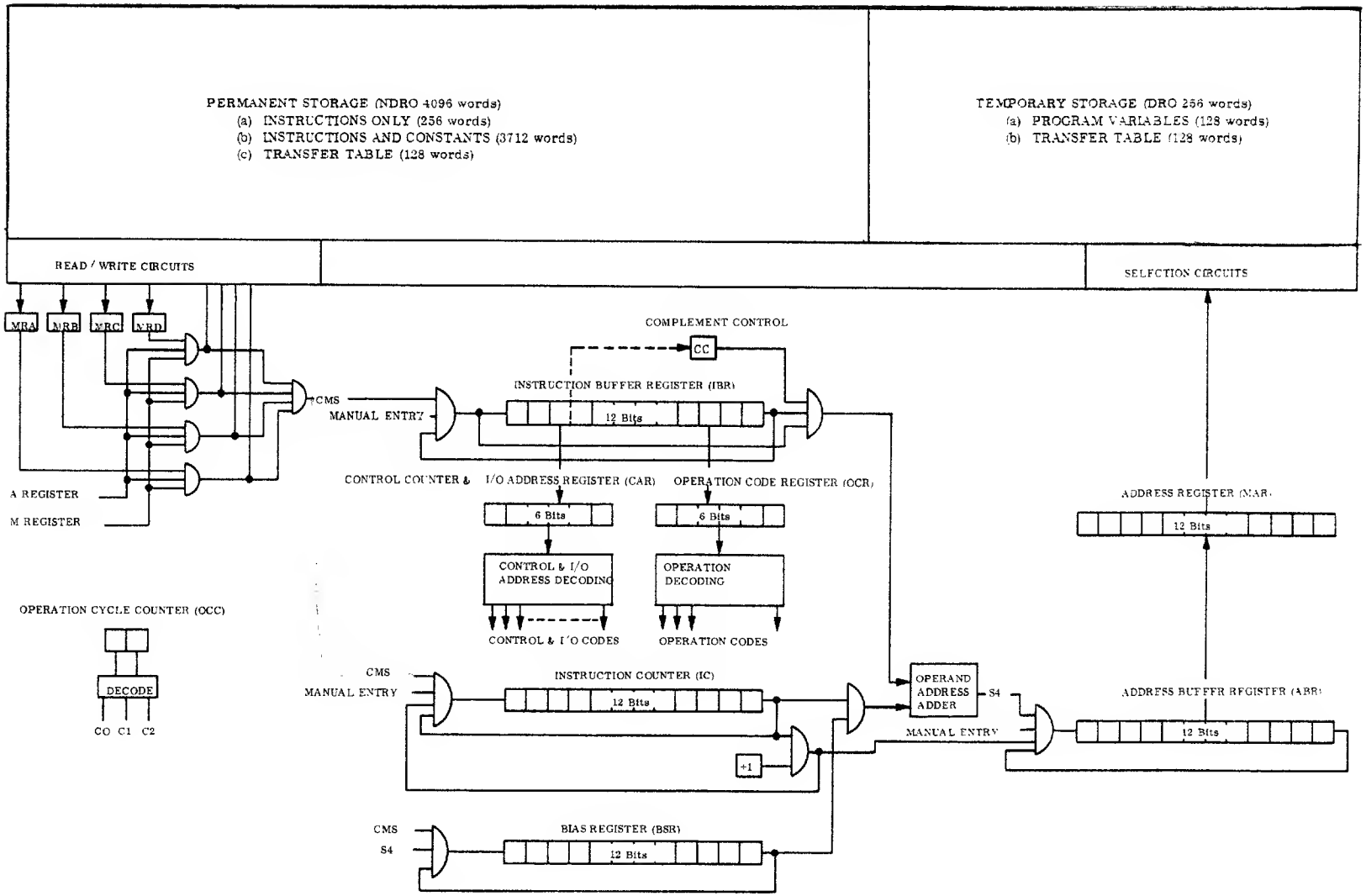


Figure 2. Instruction Processing Unit, Block Diagram

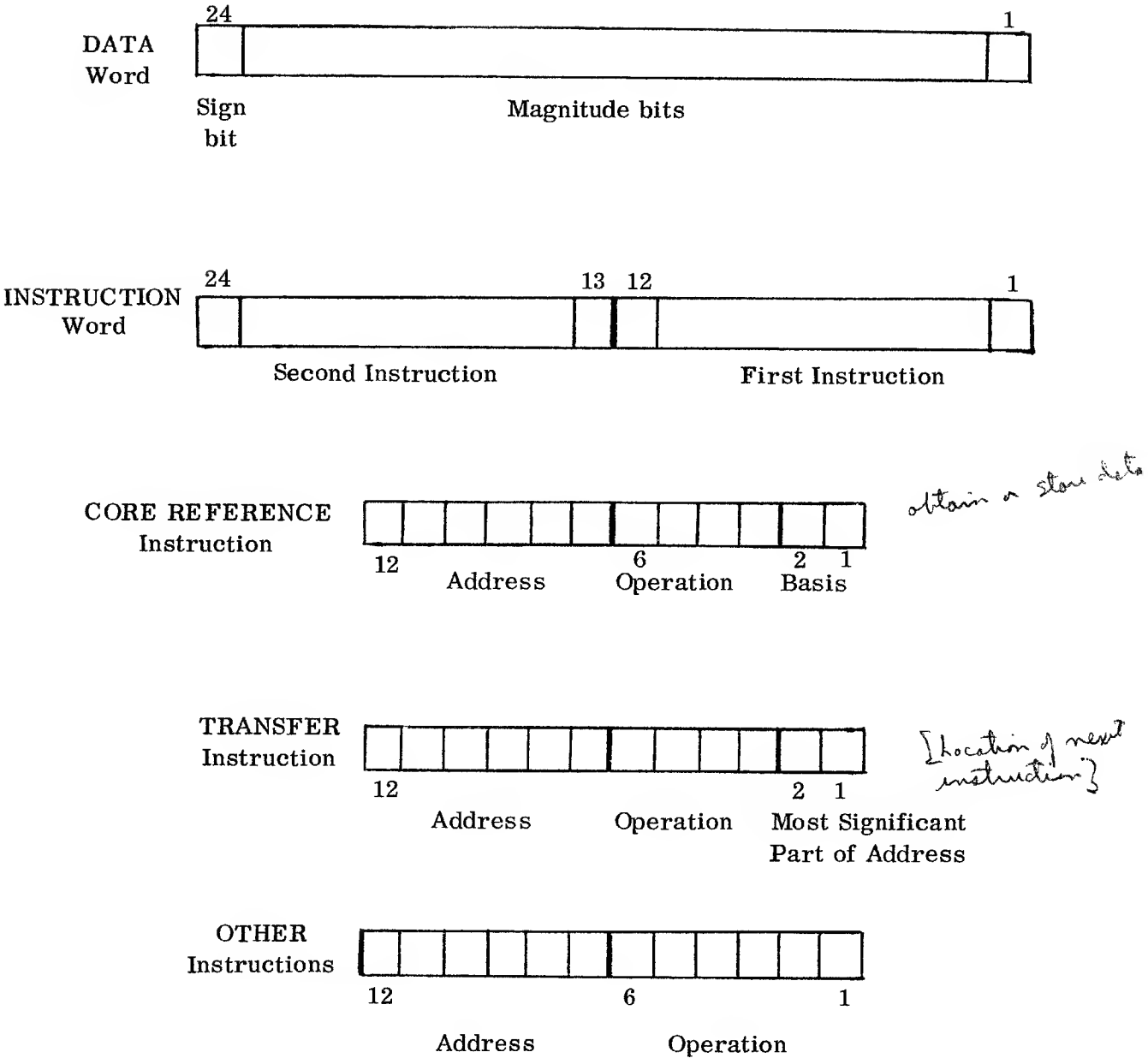


Figure 3. Data and Instruction Format

For the other transfer instruction, JOM, and for instructions that refer to the memory, relative addressing is used. For this purpose an effective address is generated by adding the bits of the address field to a reference. The instruction counter is always used as the reference for the JOM transfer instruction. In the core reference instructions, there are four references for relative addressing; two bits of the instructions, called the basis field, are used to specify which reference is to be used.

There are two fixed references which can be added to the address field to generate an effective address; these are 396_{10} and 4032_{10}. This provides a stationary block of 128 words in the DRO memory, any one of which may be randomly accessed for both reading and writing.

There are two dynamic references for addressing. These are the bias register and the instruction counter. The address field, when used for this type of addressing, is considered to have a sign and five magnitude bits, so that a range of -32 through +31 about the reference is provided. To obtain the effective address, the address field is simply added to the contents of either the bias register or the instruction counter. If the effective address lies in the ranges 0_{10} to 127_{10} or 3968_{10} to 4095_{10}, both reading and writing can occur. Otherwise only reading can take place. The format for the core reference instructions is shown in Figure 3.

The remainder of the instructions use a six-bit operation code. For the two shift instructions, the six-bit address field specifies the length of the shift to be made. For the "set bias register" instruction, the six-bit address field is used to reset the upper portion of the bias register. The six-bit address of the "modify bias register" instruction is added to the bias register. For input-output instructions, the six-bit address field specifies the input or output device that is involved. Finally, there are the "exchange" and "no-operation" instructions, for which the address field has no significance.

A number of registers are provided in the IPU for synchronization of information transfers. These include the address buffer register, the instruction buffer register, the operation code register, and the instruction counter.

For the most part, instructions are carried out in sequence under control of the instruction counter. The instruction counter is incremented by one after each instruction pair is executed. Instruction words coming from memory are transferred serially into the instruction buffer register. The instruction that occupies the 12 least-significant bits of the 24-bit instruction word is executed first. The instruction which occupies the most significant 12 bits is executed next. The low-order six bits of the instruction to be executed are transferred to the operation code register to drive the logic circuits that interpret the instruction.

Because the speed of MAGIC-series computers is principally limited by the speed of the memory, the IPU has been designed to minimize memory idle time. To this end, a cyclic three-state counter (operation cycle counter) is used for control. When the counter is in a particular state (C_0), a 24-bit word containing two 12-bit instructions is read from the memory and shifted serially into the instruction buffer register. The two instructions are executed during the other two states of the counter (C_1 and C_2).

During the interval (C_0), when the 24-bit word is being obtained from memory, the operand address for the first instruction is computed and sent to the address buffer register. While the first instruction is being executed (C_1), the operand address for the second instruction operation is computed; while the second operation is being performed (C_2), the content of the instruction counter is increased by one and sent to the address buffer register in preparation for the next instruction access.

(b) Arithmetic Unit (AU). Three 24-bit registers are used to perform the standard arithmetic and shift operations, and to provide communication with the input-output unit. Throughout the AU, information flow is two bits at a time. A two-bit adder/subtractor is the basic arithmetic element.

Information is transmitted to and from the AU serially, two bits at a time, at the rate of 2 microseconds for each pair of bits. The A register is used as an accumulator for addition and subtraction operations, each of which is completed in a single word-time of **25** microseconds. For these operations, the addend or subtrahend is transmitted directly to the adder from the memory.

Multiplication, division, and shifting operations are carried out two bits at a time, at a 1-megacycle clock rate. For multiplication, the A and B registers are used together as a double-length shift register, while the M register holds the multiplicand throughout the entire operation. The multiply control provides for the processing of two bits of the multiplier simultaneously. Multiplication time is 178 microseconds, exclusive of access time, regardless of the numbers involved. A double-length product results from multiplication.

For division, the A and B registers again act together as a double-length shift register, and the M register holds the divisor. The nonrestoring division algorithm is used to generate a 24-bit quotient and a 24-bit residue. The time required for division is 330 microseconds, regardless of the numbers involved.

The A and B registers are used together for shifting operations involving double-length numbers. Both left and right shifts can be accomplished, with the length of the shift specified in the address field of the shift instruction.

The A register also serves as a link with the input-output unit. Information transmitted to the computer for processing or storage is transferred serially from the input-output unit to the A register. Information from the computer is transmitted to the input-output unit through the A register.

(c) Memory. The memory for the MAGIC II computer requires capability for storage of both fixed and variable information. A nondestructive readout (NDRO) capability is highly desirable, in light of the possibility of temporary power interruptions or transients. Consequently, a 4096 24-bit word nondestructive readout memory is provided. Variable information is held in a 256-word (24 bits/word) memory which is modifiable by the program.

To assure high reliability without loss of critical information, the NDRO memory contains all of the program instructions and the constant data that are not required to be changed in the course of the mission. The NDRO portion of the memory uses transfluxors, and is organized such that information can be written into the memory by following a normal load procedure using the separate Computer Fill Set connected to the computer. The Computer Fill Set contains the necessary power supply and write amplifiers. Once the computer has been filled with the proper information, a verify routine can be performed to assure correct loading. The fill set may then be disconnected. With the fill set disconnected, it is physically impossible for the information in the permanent memory to be altered by either an erroneous program step or by a power transient.

Variable information is held in a "scratch pad memory," which uses toroidal ferrite cores and has a capacity of 256 24-bit words. Memory access is completely random, although information is sequentially read out serially, four bits at a time, at a 250 kc rate. The variable memory holds all temporary information as well as those quantities that must be continually updated, such as velocity and position terms. In addition, a portion of the transfer table, used for program jumps and address modification, is contained in the variable store. To reduce complexity, a common address register is used to interrogate either the permanent or the temporary memory. If instructions are being accessed, the NDRO memory will always be selected regardless of the address. If an operand address lies between 0 and 127_{10} or 3968_{10} and 4095_{10}, the DRO memory will be selected. Consequently, information from either memory is available to the programmer on a mutually exclusive basis.

The memory may also be divided according to usage. Words 0 to 127_{10} in the DRO memory may contain either transfer table entries or infrequently changing variables. Locations 3968_{10} to 4095_{10} of the DRO memory (the second 128 words) are used for storage of variables and temporary results. The first and last 128 words of the NDRO memory may only be used for instructions. Words 256_{10} to 3967_{10} are used for nonchanging constants and instructions and words 128_{10} to 255_{10} contain the permanent transfer table entries.

(d) Input-Output Processing Unit (IOU). The IOU for MAGIC II was designed to fulfill requirements of a particular avionics system and thus contains the capability to interface between the subsystems itemized in Table 2 and the computer program. This feature of MAGIC II will not be discussed in detail due to its highly specialized nature.

Briefly, the MAGIC II IOU includes input facilities for handling the following types of inputs: (1) incremental pulses on positive and negative lines, (2) phase-variable, square-wave inputs, (3) whole-number values from shaft encoders, and (4) discrete or "on-off" signals. Output capabilities include generation of variable-frequency and variable-width pulse trains, and discrete signals.

An additional function of the input/output unit is the timing of the interrupt commands sent to the IPU. A signal is sent every 40 milliseconds to the IPU to interrupt the program. The effect of this signal is such that, at the completion of the instruction pair presently being executed, the instructions in location 128 will be executed. If only nontransfer type instructions are present, control will be sent back to the point from which it came, and the program will continue its normal execution. If a transfer instruction is present in cell 128, it will be executed in the normal manner. In normal usage, where a high-speed loop must be performed many times a second in relation to the main program, a Transfer to Subroutine (TRS) will be coded in location 128. This causes the contents of the location counter and the bias register to be trapped in location 1 of the variable memory. The last instruction in the high-speed loop is an indirect transfer to location 1 returning control to the main program at the correct point. Additional interrupts may be provided on the basis of data availability or other such criteria.

SECTION III

MAGIC ORDER CODE

The detailed order code configuration for the MAGIC II computer is described below. Mnemonic codes, execution times, and descriptions of the operations are given in three categories: those requiring core references; transfer operations; and others. The execution times include memory access times for both the instruction and the operand, based on a 1.024 Mc clock rate. Table 3 presents a summary of the order code.

Mnemonic Code	Time (μs)	Memory Reference Operations
LDA	38	Load A Register from Memory. The contents of the A register are replaced by the contents of the effective address[1]. The contents of the effective address are unchanged.
STO	38	Store A Register into Memory. The contents of the effective address are replaced by the contents of the A register. The contents of A are unchanged.
ADD	38	Add to A Register. The contents of the effective address are added algebraically to the contents of the A register, and the sum is placed in the A register. The process involves normal binary addition, negative numbers being represented in two's complement form. The sign position of the A register inverts upon overflow.
SUB	38	Subtract from A Register. The contents of the effective address are subtracted algebraically from the contents of the A register, and the difference is placed in the A register. The sign position of the A register inverts upon overflow.

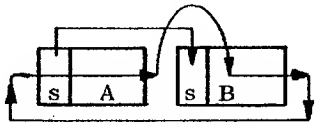
[1]The term "effective address" refers to the address transferred into the memory address register after modification of the address field by the bias register, instruction counter, or other fixed biases.

Mnemonic Code	Time (μs)	Memory Reference Operations (cont)
MPY	190	Multiply. The contents of the effective address are multiplied algebraically by the contents of the A register. The 48-bit product replaces the contents of the combined A and B registers. The most-significant half of the product is in the A register; the least-significant half is in the B register. The sign of the product is in the sign positions of both the A and B registers.
DIV	342	Divide. The contents of the combined A and B registers are divided algebraically by the contents of the effective address. The sign of the A register is the sign of the dividend, the sign of the B being ignored. The quotient replaces the contents of the A register, while the residue replaces the contents of the B register. The dividend must be less in absolute value than the divisor, or overflow will occur.
MSK	38	Mask A Register. The contents of the A register are logically multiplied by the contents of the effective address, and the logical product replaces the contents of the A register. If any bit position of the contents of the effective address contains a zero bit, the corresponding bit position of the A register is replaced by zero. All other bit positions of the A register are left unchanged.
		Transfer Operations
TRA	51;38	Transfer Unconditionally. The next program instruction will be taken from the first segment of the location specified by the 12 least-significant bits of a word in the transfer table. The address portion of the TRA instruction designates the transfer table word. The bias register is loaded with the 12 most-significant bits of the transfer table word. If this TRA command is in the first segment of the instruction, 51 μs are required for execution; otherwise 38 μs are required.
TRM	51;38	Transfer on Minus A Register. If a "1" is contained in the sign position of the A register, a TRA is executed. If the sign position is "0" the program proceeds to the next operation in sequence. A 38μs period is required for an unexecuted transfer.

Mnemonic Code	Time (μs)	Transfer Operations (cont)
TRZ	51;38	Transfer on Zero A Register. The contents of the A register are examined for zero after the execution of the operations LDA, STO, ADD, SUB, MSK, INP, OUT, DSI, and XAB. If zero is present and a TRZ is then encountered, a TRA will be executed. Otherwise the program will proceed to the next operation in sequence. TRZ is not valid if it immediately follows a MPY, DIV, LRS, or RTE without an intervening instruction from the above set. 38μs are required if the transfer is unexecuted.
JOM	51;38	Jump on Minus. A jump forward or backward the number of locations specified by the address portion of this instruction is done if the sign position of the A register contains "1". Jumps backward are caused by placing the two's complement of the number of locations desired in the address field. The jumps range from -128 to +127 locations. If the JOM is coded as the first segment instruction, 51 μs are required for execution; otherwise 38 μs are required.
TRS	76;63	Transfer to Subroutine. The contents of location 0000 (first entry in transfer table) are replaced by the contents of the bias register and the instruction counter; the bias register going to the most-significant 12 bits, and the instruction counter going to the least-significant 12 bits. A TRA is then executed. The instruction counter contents are the current instruction address plus one. If the interrupt condition is present, the instruction counter and bias register will be placed in location 0001 instead of location 0000. If the TRS is coded in the first segment of the instruction word, 76 μs are required. Otherwise 63 μs are required.
		Input-Output Operations
INP	38	Input. The contents of the A register are replaced by the contents of the I/O register specified by the address portion of this instruction. Usually the I/O register is cleared.
OUT	38	Output. The contents of the I/O register specified by the address portion of this instruction are replaced by the contents of the A register. The contents of the A register are unchanged.

Mnemonic Code	Time (μs)	Input-Output Operations (cont)
DSI	38	Discrete Input. The sign position of the A register is set to "0" or "1" according to the absence or presence, respectively, of the discrete signal specified by the address portion of this instruction. All other bits of the A register remain the same as they were.
DSO	38	Discrete Output. A discrete signal is sent to the device specified by the address portion of this instruction.
		Miscellaneous Operations
LRS	38;51	Long Right Shift. The contents of the A and B registers combined are shifted right by the number of places specified by the address portion of this instruction. Bits shifted past the least-significant bit of the A register enter the most-significant nonsign bit of B. Bits shifted past the least-significant bit of B are lost. The sign of the A register is copied into both the most-significant bit positions of A and the sign position of B as the number is shifted. If the number of positions to be shifted is one, or is even, 38μs are required. If the number of positions to be shifted is greater than two, and is odd, 51μs are required.
RTE	38;51	Rotate Right. The contents of the combined A and B registers are rotated right by the number of places specified by the address portion of this instruction. An effective long left shift is achieved by rotating right 47 minus the number of places desired to shift left. Bits shifted past the least-significant bit of B enter the sign position of A. After shifting is complete, the new sign of A is copied into the sign of B. If the number of positions to be shifted is one, or is even, 38μs are required. If the number of positions to be shifted is greater than two, and is odd, 51μs are required.
XAB	38	Exchange A and B. The contents of the A and B registers are interchanged.
MBR	38	Modify Bias Register. The address field is treated as a 6-bit two's complement number, and it is added algebraically to the low-order six bits of the bias register.

Mnemonic Code	Time (μs)	Miscellaneous Operations (cont)
SBR	38	Set Bias Register. The high-order six bits of the bias register are replaced by the six-bit address field of this command. The low-order six bits of the bias register are reset to zero.
NOP	38	No Operation. The next instruction in the normal sequence of instructions is executed.

CATEGORY	OPERATION	TIME (μs)	NMEMONIC CODE	DESCRIPTION
CORE REFERENCE	Load A from memory	38	LDA	(Z) → (A)
	Store A into memory	38	STO	(A) → (Z)
	Add	38	ADD	(A)+(Z) → (A)
	Subtract	38	SUB	(A)-(Z) → (A)
	Multiply	190	MPY	(A)· (Z) → (AB)
	Divide	342	DIV	(AB) ÷ (Z); Quot → (A); Res → (B)
	Mask	38	MSK	Logical product of (A) and (Z) → (A)
TRANSFER	Transfer	51/38	TRA	(Z) → (IC), (BR)
	Transfer on Minus A	51/38	TRM	(Z) → (IC), (BR) if A_S is "1"
	Transfer on Zero A	51/38	TRZ	(Z) → (IC), (BR) if (A)=0
	Transfer to Subroutine	76/63	TRS	(IC), (BR) → (0000) or (0001) (Z) → (IC), (BR)
	Jump on Minus	38	JOM	(IC)+IA → (IC)
OTHER	Long Right Shift	38/51	LRS	
	Rotate Right End Around	38/51	RTE	
	Exchange A and B	38	XAB	(A) → (B); (B) → (A)
	Modify Bias	38	MBR	BR + IA → BR
	Set Bias	38	SBR	IA → $(BR)_{msh}$; 0 → $(BR)_{lsh}$
	No Operation	38	NOP	
INPUT/ OUTPUT	Input	38	INP	(Input) → (A)
	Output	38	OUT	(A) → (Output)
	Discrete Input	38	DSI	Signal → (A_S)
	Discrete Output	38	DSO	Signal → Interface

KEY:
- A = A Register
- A_S = Sign position of A Register
- B = B Register
- BR = Bias Register
- IA = Address portion of current instruction
- IC = Instruction Counter
- (X) = Contents of X
- Z = Designated core memory word
- → = Replaces
- msh = most significant half
- lsh = least significant half

Table 3. Summary of Order Code